ADSP Home-work 1: MIMO System Identification and Deconvolution

Gabriele Bruni

July 2022

1 Noise generation

1.1 Generating the noise

Noise is Gaussian, temporally white, and mutually correlated among paired lines, i.e

$$\mathbb{E}[\mathbf{w}[k]\mathbf{w}[l]] = \mathbf{C}\delta[l-k]$$

$$\mathbf{w}[k] \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$$

where

$$[\mathbf{C}]_{i,j} = \begin{cases} \rho & i \neq j \\ 1 & i = j \end{cases}$$

with $|\rho| \neq 1$.

For a linear transformation $\mathbf{w} = \mathbf{A}\mathbf{v}$ of a Gaussian random variables $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the mean vector and the covariance matrix will result respectively in

$$\boldsymbol{\mu}_w = \mathbf{0}$$

$$\mathbf{C}_{ww} = \mathbf{A}\mathbf{A}^T.$$

By doing the Cholesky factorization of our desired covariance matrix, we get a matrix that can be used as A in the above equation to color our noise.

In Matlab this is accomplished first generating a white noise vector with the function `randm(M,P)` and then correlating it using `chol(A).` Both steps can be collapsed using the built-in function `mvnrnd(MU,SIGMA,N)` (that returns a N-by-D matrix of random vectors chosen from the multivariate normal distribution with 1-by-D mean vector MU, and D-by-D covariance matrix SIGMA. This will be the default choiche for the next exercises).

1.2 Estimating the covariance matrix

Estimating of covariance matrix is done by using the `cov(W)` Matlab built-in function which returns the covariance of a matrix, where each row is an observation. The resulting estimation is then evaluated with respect the original covariance matrix using the MSE metric, avareged using a Montecarlo approach. Figure 1 shows the MSE of the estimate as function of L for different value

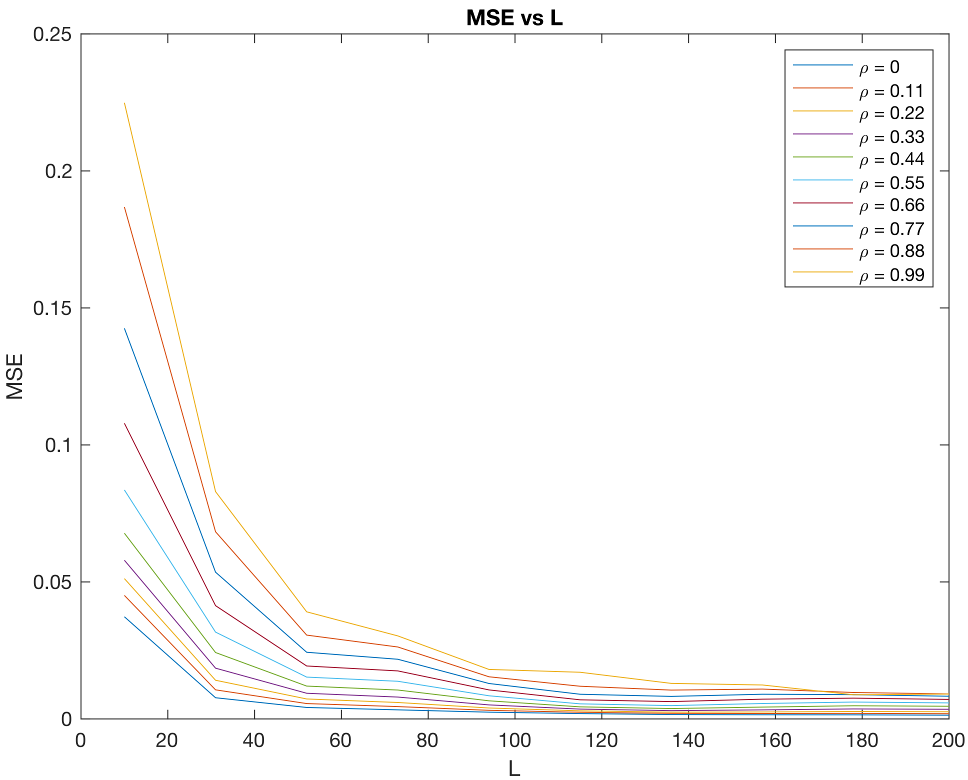

Figure 1: Comparison between the estimate of the covariance matrix and the actual covariance matrix, as function of L.

of ρ, while figure 2 shows the MSE of the estimate as function of ρ for different value of L. We can see that the estimate improves as L increases (we have more data to calculate our estimate) and get worse when ρ increase.

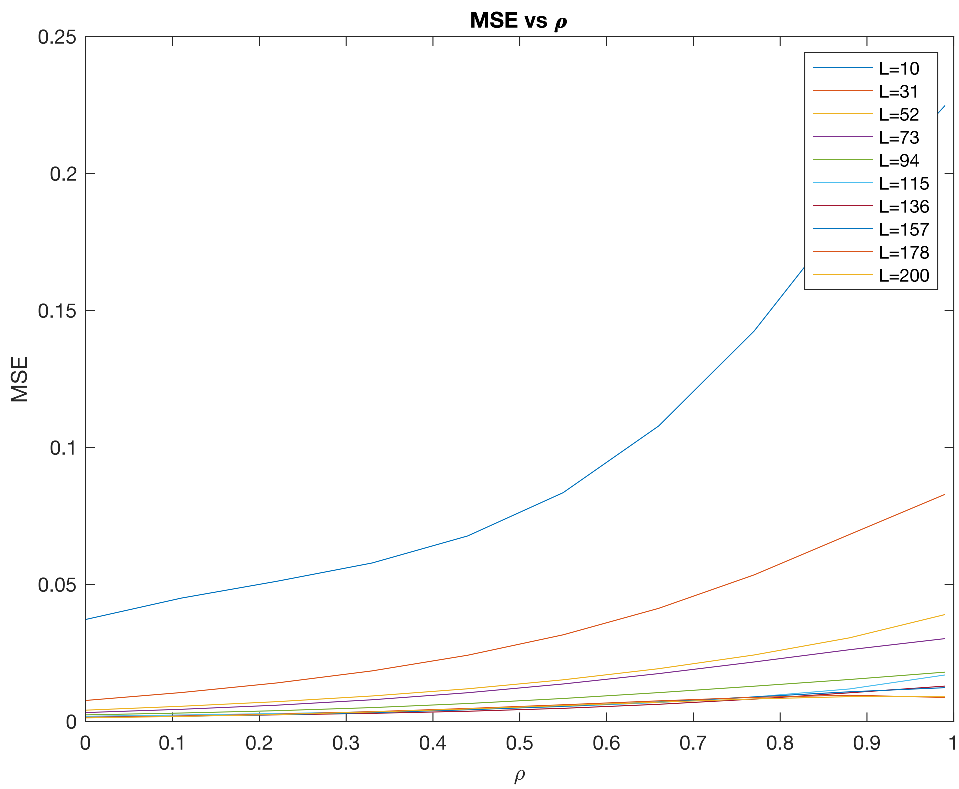

Figure 2: Comparison between the estimate of the covariance matrix and the actual covariance matrix, as function of ρ.

2 MIMO estimation

2.1 Defining the MLE

Normally, the system is represented as

$$\mathbf{y}[k] = \mathbf{H}[k] * \mathbf{x}[k] + \mathbf{w}[k]$$

In order to evalute the MLE, we rewrite the system as

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}$$

where $\mathbf{X}$ is the convolution matrix of the input signal $\mathbf{x}$, $\mathbf{h}$ is the filter response in vector form, and $\mathbf{w}$ is the correctly arranged noise vector. We define the MLE of $\mathbf{h}$ as

$$\hat{\mathbf{h}}_{ML} = (\mathbf{X}^T \mathbf{C}_w^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{C}_w^{-1} \mathbf{y}$$

We define the convolution matrix $\mathbf{X}$ in three steps: First, for each of the N input vector $\mathbf{x}_i = [x_i[0], \ldots, x_i[Q-1]]$ we build the matrix

$$\mathbf{X}_i = \begin{bmatrix} x_1[1] & 0 & & 0 \\ \vdots & x_1[1] & & \vdots \\ x_1[Q-1] & \vdots & \ddots & x_1[1] \\ 0 & x_1[Q-1] & \ddots & \vdots \\ 0 & 0 & & x_1[Q-1] \end{bmatrix}$$

with dimension of $Q + K - 1 \times K$. In the memoryless case $K = 1$, this will collapse in a single column vector. Then we stack those N matrices

$$\mathbf{X}_{conv} = \begin{bmatrix} \mathbf{X}_1 & \ldots & \mathbf{X}_N \end{bmatrix}$$

obtaining a matrix of dimension $Q + K - 1 \times NK$. Finally, we define our convolution matrix as

$$\mathbf{X} = \mathbf{X}_{conv} \otimes \mathbf{I}_{M \times M}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \ldots & \mathbf{X}_N & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{X}_1 & \ldots & \mathbf{X}_N & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & & \ddots & & & & & & \vdots & \\ \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{X}_1 & \ldots & \mathbf{X}_N \end{bmatrix}$$

where $\mathbf{I}_{M \times M}$ is the $M \times M$ identity matrix and $\otimes$ is the Kronecker product, obtaining a $M(Q + K - 1) \times MNK$ matrix.

To generate $\mathbf{y}$, we define the filter response as a $MNK \times 1$ column vector $\mathbf{h}$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \ldots & \mathbf{h}_{MN} \end{bmatrix}^T$$

where

$$\mathbf{h}_{ij} = \begin{bmatrix} h_{ij}[0] & \ldots & h_{ij}[K-1] \end{bmatrix}^T$$

,

and the noise as a $M(Q+K-1) \times 1$ vector, $\mathbf{w}$

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 & \dots & \mathbf{w}_M \end{bmatrix}^T$$

where

$$\mathbf{w}_i = \begin{bmatrix} w_i[0] & \dots & w_i[Q+K-1] \end{bmatrix}^T$$

The covariance generating the noise is defiend as a $M(Q*K-1) \times M(Q+K-1)$ matrix $\mathbf{C}_w$

$$\mathbf{C}_w = \mathbf{C} \otimes \mathbf{I}_{Q+K-1}$$

to preserve the spatial correlation. The error with respect to the true filter is evaluated as

$$\mathbf{err} = \hat{\mathbf{h}}_{ml} - \mathbf{h}$$

Since $\mathbf{y}$ is drawn from an additive Gaussian model with zero-mean noise, the Cramer-Rao bound, which represent the theoretical lower bound for the MSE reachable by the MLE estimator, is defined as

$$\mathbf{CRB} = (\mathbf{X}^T \mathbf{C}_w^{-1} \mathbf{X})^{-1}$$

It has to be noted that CRB is a matrix, but a scalar is needed to be compared with the MSE. Therefore, the average of the diagonal entries of CRB matrix is considered as lower bound value.

2.2 Memoryless filter

From both figure we can note that estimation improve as SNR is increased. From figure 3 we note that more Q samples lead also to better channel estimation in terms of MSE. From figure 4 we can also see that different values of α do not affect the perfromance of the MSE.

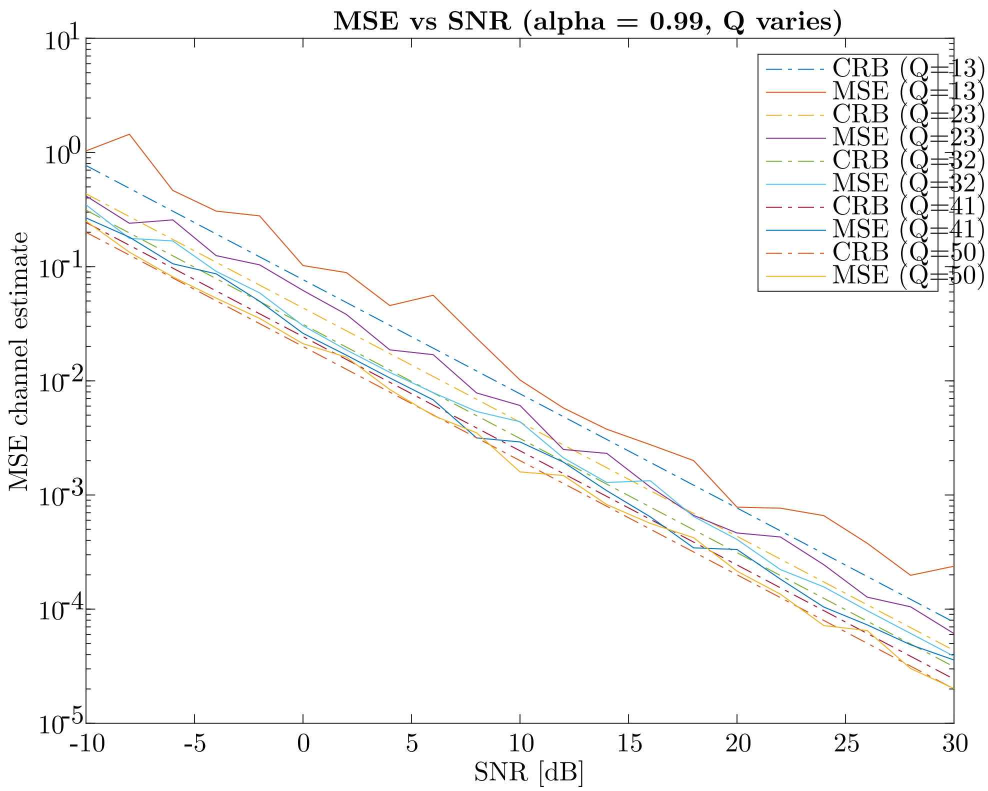

Figure 3: MSE vs SNR, α fixed, Q varies.

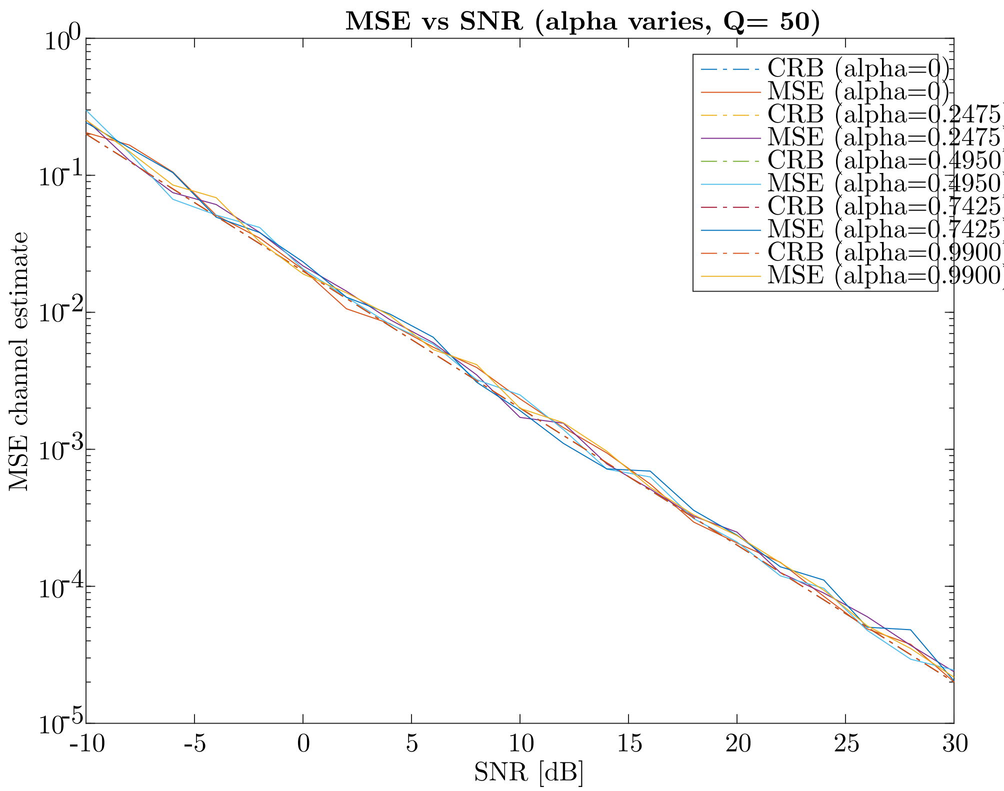

Figure 4: MSE vs SNR, α varies, Q fixed.

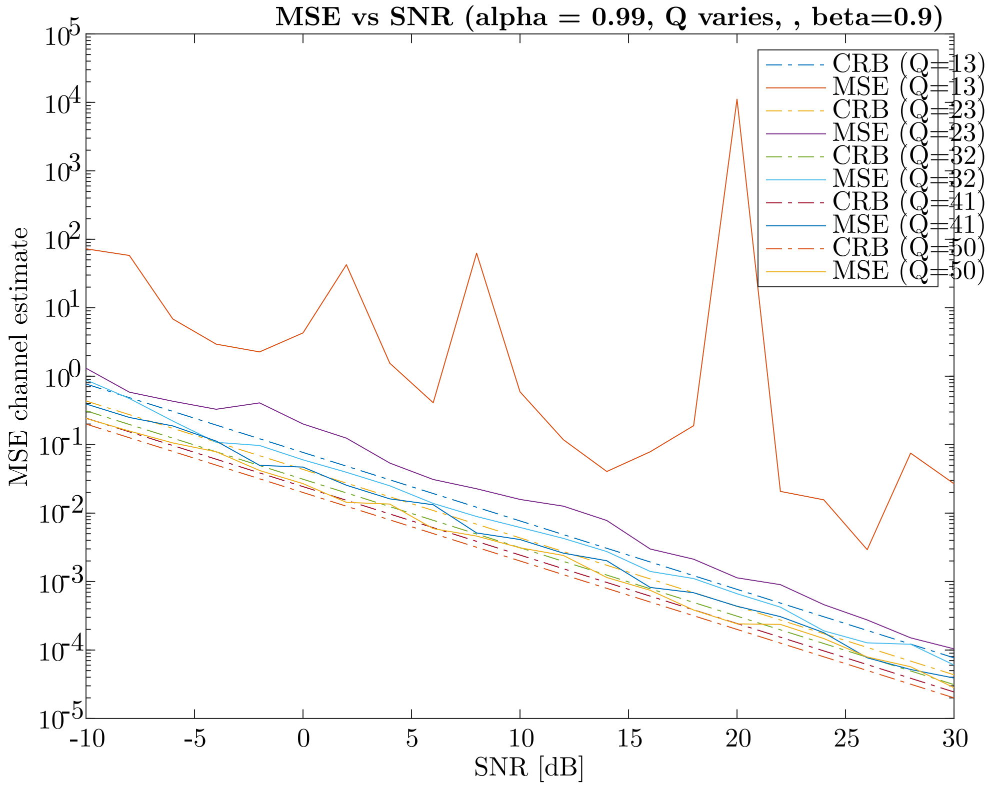

Figure 5: MSE vs SNR, α fixed, Q varies, beta fixed.

2.3 Memory filter

The same result apply to memory filter. Here we have also the β parameter. And also here we can see that different values of β parameter does not affect the performance of the estimator (see figure 6).

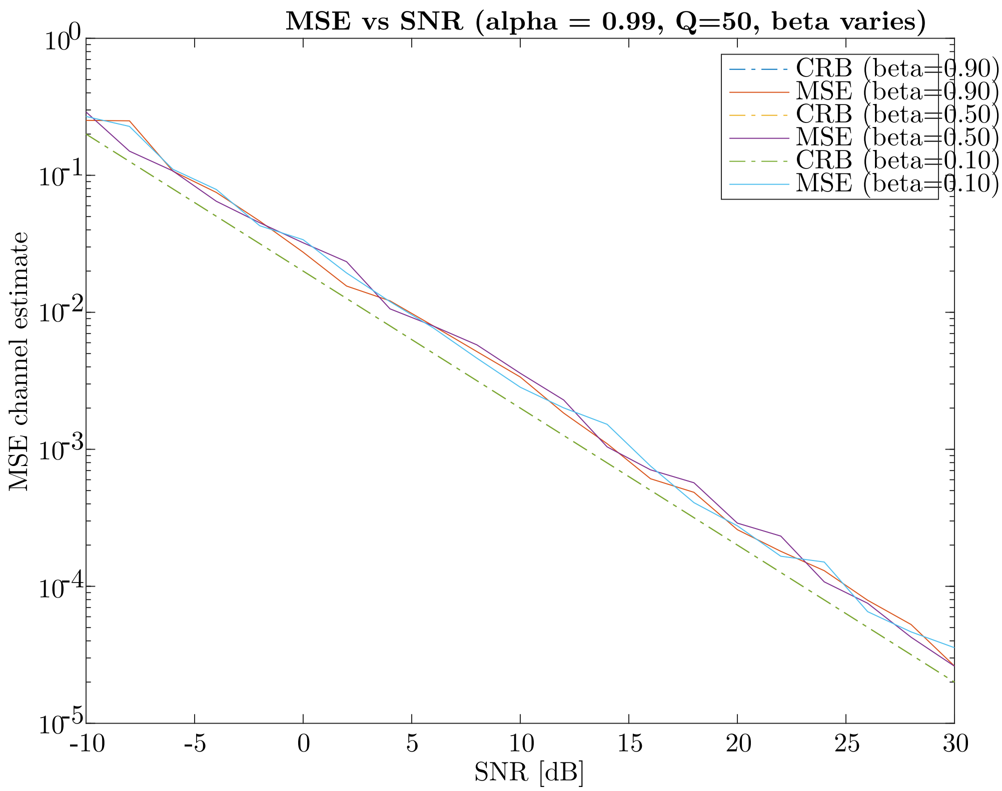

Figure 6: MSE vs SNR, α fixed, Q fixed, beta varies.

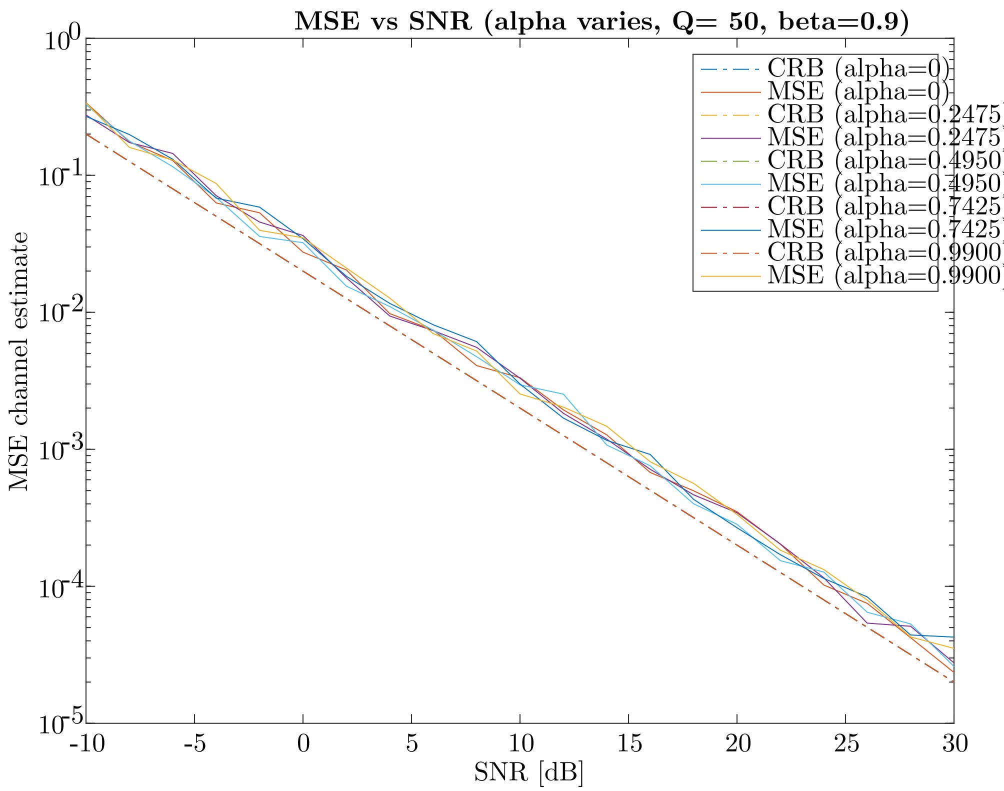

Figure 7: MSE vs SNR, α varies, Q fixed, beta fixed.

3 MIMO deconvolution

First we generate the transmitted data $\mathbf{x}_Q$ and then we compute the MLE estimate the filter $\hat{\mathbf{h}}_{mle}$ through the procedure illustred previously. Then we estimate the remaining data as

$$\mathbf{y}_P = \hat{\mathbf{H}}\mathbf{x}_P + \mathbf{w}$$

Where $\hat{\mathbf{H}}$ is the convolution matrix obtained from $\hat{\mathbf{h}}_{mle}$. Then we compute the MLE and MMSE of $\mathbf{x}$ as

$$\mathbf{x}_{mmse} = \boldsymbol{\mu}_x + (\hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \hat{\mathbf{H}} + \mathbf{C}_x^{-1})^{-1} \hat{\mathbf{H}}^T \mathbf{C}_w^{-1} (\mathbf{x}_P - \hat{\mathbf{H}}\boldsymbol{\mu}_x)$$

$$\mathbf{x}_{mle} = \ (\hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \hat{\mathbf{H}})^{-1} \hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \mathbf{x}_P$$

The difference between the MLE and the MMSE is that the MLE uses only a- posteriori information, while the MMSE uses both a-priori and a-posteriori information. Since $\mathbf{x}_P$ is zero-mean the MMSE becomes

$$\mathbf{x}_{mmse} = (\hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \hat{\mathbf{H}} + \mathbf{C}_x^{-1})^{-1} \hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \mathbf{x}_P$$

Comparaing this to MLE, we can see that the only difference is the addition of $\mathbf{C}_x^{-1}$ term. Since $\mathbf{C}_x = \mathbf{I}\boldsymbol{\sigma}_x^2$ its inverse will be very small when SNR is large. So as SNR grows, the MMSE estimator should approach the MLE estimator. If SNR is very small then the inverse of $\mathbf{C}_x$ should grow very big, such that

$$\mathbf{x}_{mmse} = (\mathbf{C}_x^{-1})^{-1} \hat{\mathbf{H}}^T \mathbf{C}_w^{-1} \mathbf{x}_P$$

If the SNR is very high, then the noise in our received signal is negligible, and we don't need to use a-priori information in the estimation, hence MMSE approaches the MLE. On the other hand, with a very low SNR, the received signal is becoming useless, so it can be neglected. Thus only the a-priori information is used, so MMSE outperform the MLE alternative.

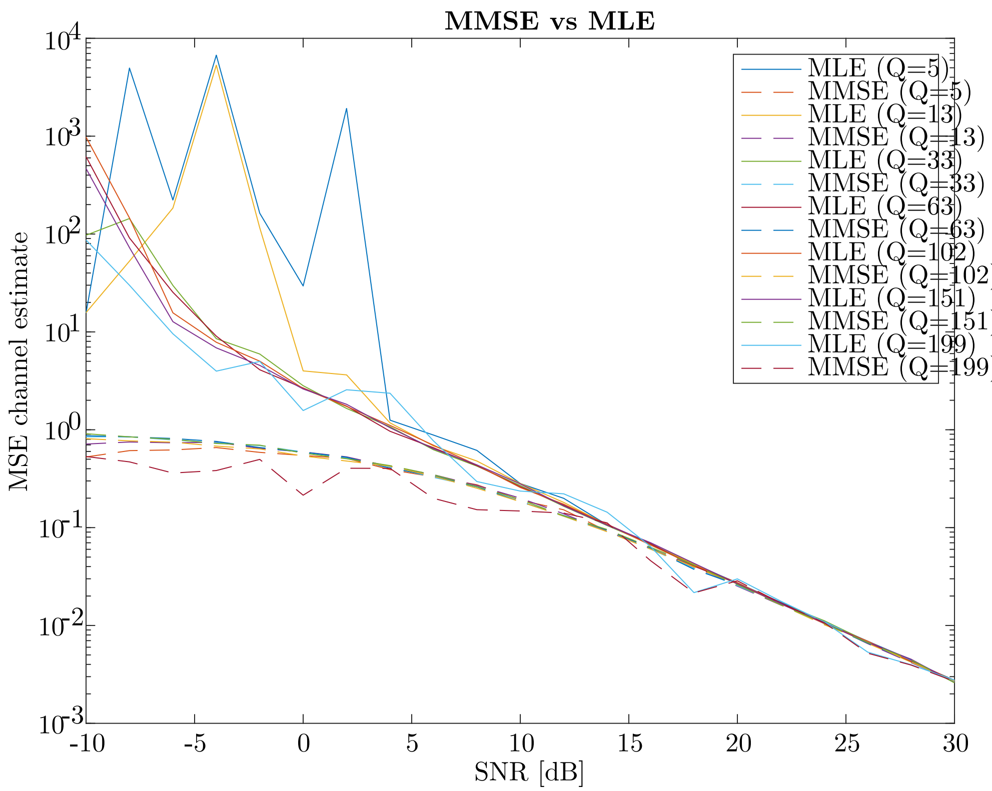

Figure 8: MMSE vs MLE, Q varies.